

NintendoWare for Cafe

Sound System Manual - Sound System Operating Specifications

2015/07/03

Version 1.0.2

**The content of this document is highly confidential
and should be handled accordingly.**

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo and/or its licensed developers and are protected by national and international copyright laws. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

Table of Contents

1	Introduction	6
1.1	About This Manual	6
2	Binary File Structure	7
2.1	Sound Archive File	7
2.1.1	Items.....	7
2.1.2	Reference File Path.....	7
2.2	Stream File.....	7
2.2.1	SoundMaker Output Destination and Location in Storage.....	7
3	3D Sound	8
3.1	Description of Terms	8
3.1.1	3D Sound Listener.....	8
3.1.2	3D Sound Actor	8
3.1.3	Interior Size	8
3.1.4	Maximum Sound Volume Range.....	8
3.1.5	Attenuation Unit Distance.....	8
3.1.6	Maximum Priority Reduction	8
3.2	Operational Specifications	9
3.2.1	Library Operations.....	9
3.2.2	SoundMaker Operations	12
3.3	Interior Size and Maximum Sound Volume Range.....	12
3.3.1	Relationship Between the Interior Size and Maximum Sound Volume Range	12
3.3.2	Interior Size Properties.....	13
4	Priority Behavior.....	16
4.1	Types of Priority	16
4.1.1	Player Priority	16
4.1.2	Voicing Priority.....	16
4.2	Operation Specifications	16
4.2.1	Player Priority	16
4.2.2	Voicing Priority.....	17
4.2.3	Determining Priority When Several Targets Are Assessed Under the Same Conditions.....	18
4.2.4	Special Priority Determination Processing	18
5	Pan Operations	19
5.1	Description of Terms	19
5.1.1	Pan Mode	19
5.1.2	Balance Pan	19

5.1.3	Dual Pan	19
5.1.4	Pan Curve	19
5.2	Operational Specification	19
5.2.1	Pan Operation on Stereo Waveforms	19
5.2.2	Pan Mode Settings.....	20
5.2.3	Pan Curve Types, Pan Curve Settings	20
6	Sound Parameter Control Operations	22
6.1	Setting and Controlling Sound Parameters	22
6.2	Operating Specifications.....	22
6.2.1	Multiplication	22
6.2.2	Addition	23
6.2.3	Priority	23
7	Number of Sound Instances	24
7.1	Relationship to Memory Size.....	24
7.2	Number of Sounds.....	25
7.2.1	Comparison with Player Sound Limits	25
7.3	Number of Sequence Tracks	25
7.4	Number of Stream Channels	26
8	Migration Guide	28
8.1	Migrating from NintendoWare for Revolution	28
8.1.1	Import Methods	28
8.1.2	Features Revised and Parameters Added/Removed When Importing.....	28
8.1.3	Changes to the Sound Runtime.....	29
8.2	Migrating from NITRO-System and TWL-System	29
8.2.1	Import Methods	29
8.2.2	Features Revised and Parameters Added/Removed When Importing.....	30
9	Troubleshooting	32
9.1	StartSound Succeeds but Sound Does Not Play	32
9.2	Certain Notes Do Not Play in Sequence Sounds	32
	Revision History	33

Tables

Table 5-1 List of Pan Modes That Can Be Set with SoundMaker.....	20
Table 5-2 List of Pan Curves That Can Be Set with SoundMaker.....	20
Table 6-1 Sound Parameter Types and Corresponding Control Methods.....	23
Table 8-1 Relationships Between the Original and Changed File Extensions.....	29

Figures

Figure 3-1 A Listener and an Actor	9
Figure 3-2 3D Sound Schematic 2.....	10
Figure 3-3 Example When the Attenuation Unit Distance Is Set to 1000	11
Figure 3-4 Interior Size and Maximum Sound Volume Range	13
Figure 3-5 Interior Size Property 1.....	14
Figure 3-6 Interior Size Property 2.....	15
Figure 5-1 Pan Curve Types	21
Figure 6-1 Sound Parameter Control.....	22
Figure 9-1 Region Where Waveform File Is Not Assigned	32

1 Introduction

This overview describes the operational specifications of the sound development environment provided by NintendoWare for Cafe (hereafter referred to as "NintendoWare").

1.1 About This Manual

This document was created for sound programmers, sound designers, composers, and other users of the NintendoWare sound development environment. It defines terms and explains operational concepts and operations specifications that can be performed with SoundMaker and the sound library (`nw:snd`). Features that embody new or complex concepts are emphasized.

For basic sound specifications, see the separately distributed *Sound Programming Guide*.

2 Binary File Structure

This chapter describes the files that are written to storage for use by the `nw:snd` library when the application software is packaged for distribution.

2.1 Sound Archive File

The sound archive file, which is created by SoundMaker and given the extension `.bfsar`, contains the sound data used by the `nw:snd` library. In addition to the information about players, groups, and waveform archives; the file contains data for stream, wave, and sequence sounds, the sequence data and banks, and the waveform data for wave sounds. The waveform data for stream sounds is not included and is instead referenced from an external stream file (see Section 2.2 Stream File). The file can be stored anywhere in storage.

2.1.1 Items

The sound archive file is comprised of such items as stream sounds, wave sounds, sequence sounds, banks, waveform archives, players, and groups. All items have string names and can total 2^{32} (approximately 4.3 billion) in one file.

2.1.2 Reference File Path

To play stream sounds, the sound archive file references the stream file externally, using the relative path from the directory where the sound archive file is located.

2.2 Stream File

The stream file, which is created by SoundMaker and given the extension `.bfstm`, holds the waveform data for the stream sounds used by the sound archive file. The stream file must be saved to a location in storage where it can be referenced using the file path stored in the sound archive file.

2.2.1 SoundMaker Output Destination and Location in Storage

The stream file created by SoundMaker is normally output to the `output` directory, which is located at the same hierarchical level as the Sound Project file being converted. The SoundMaker project settings allow you to save the output stream file anywhere in the `output` directory; the location is stored in the sound archive file as the file path of the stream file. All files in `output` can be placed anywhere in storage and used as is.

3 3D Sound

This chapter explains the operational specifications for 3D Sound.

3.1 Description of Terms

This section describes the terminology used in the 3D sound field.

3.1.1 3D Sound Listener

The 3D sound listener is the conceptual equivalent of a microphone or a human ear. The 3D sound listener stores the coordinates and direction of the sounds emitted by 3D sound actors. In this document, the term *listener* refers to *sound listener*.

3.1.2 3D Sound Actor

The 3D sound actor is the conceptual equivalent of the real-world sound sources, such as people and automobiles. The 3D sound actor stores the position of and the sound generated by the sound source. In this document, *actor* refers to *sound actor*.

3.1.3 Interior Size

The interior size represents the size of the space where the positional relationship between the listener and the actor is expressed using the output balance between each speaker (variations in pan and surround pan).

3.1.4 Maximum Sound Volume Range

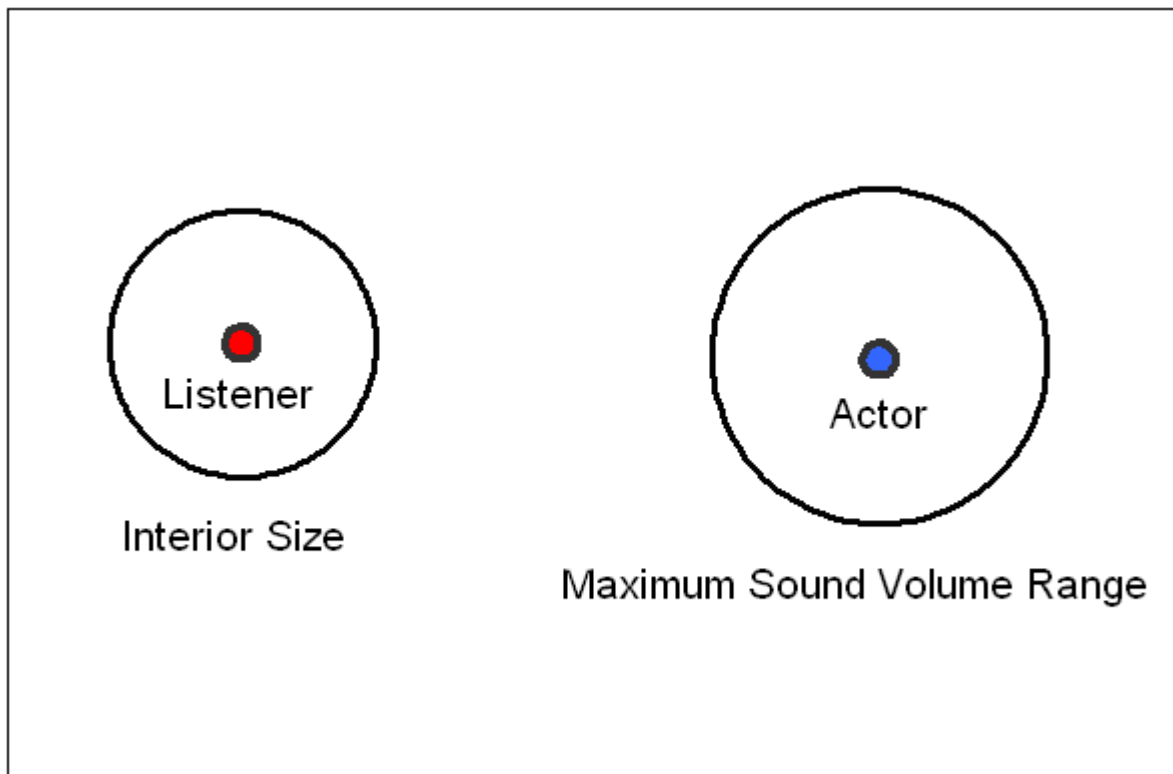
This term refers to the distance from the listener at which the sound requested by the actor is always played at maximum volume.

3.1.5 Attenuation Unit Distance

The volume and the voice priority of a sound requested by the actor attenuate in relation to the distance from the listener. The attenuation unit distance refers to the baseline distance of this attenuation.

3.1.6 Maximum Priority Reduction

The restriction on the reduction of voice priority is called Maximum Priority Reduction. It defines the extent to which the voice priority, originally set for a sound, can be attenuated.

Figure 3-1 A Listener and an Actor

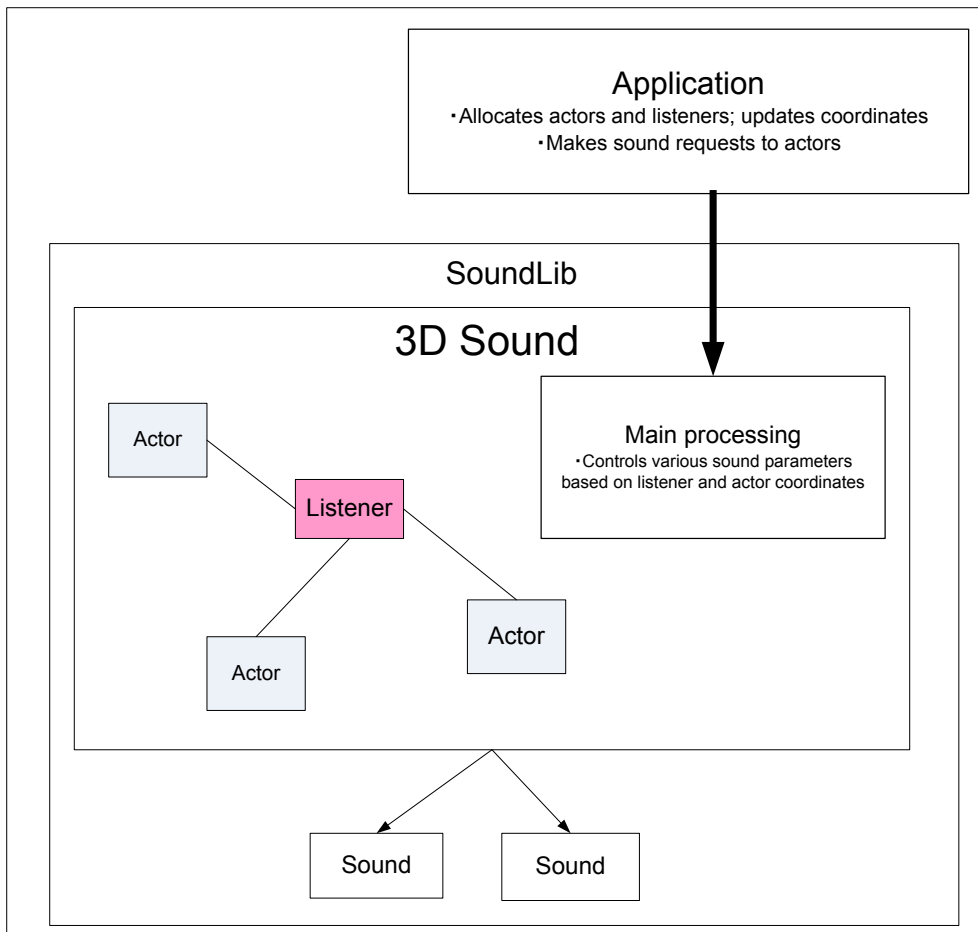
3.2 Operational Specifications

This section describes the two components of 3D sound system operations: the library and the SoundMaker. For more detailed operating instructions, see the *SoundMaker Manual* and other relevant documentation.

To read about the internal processing and priorities used to control the various sound parameters like volume, pitch, and pan set using SoundMaker and the sound library, see Chapter 6. Sound Parameter Control Operations in this document.

3.2.1 Library Operations

To apply the 3D effect to a sound, first allocate the actors and then make requests of them. By letting the application update the coordinates for the listener and actors as needed, the 3D sound system creates a simulated sound field in the speakers and sets proper parameters based on the given coordinates.

Figure 3-2 3D Sound Schematic 2**3.2.1.1 Listener and Actor Control**

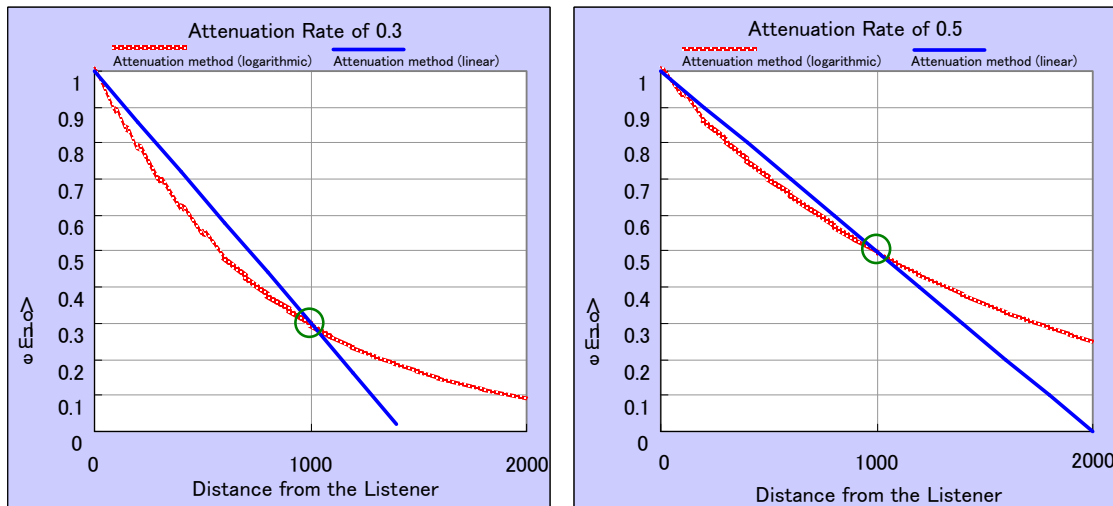
Use the API to adjust the coordinate and direction parameters of the listeners, and the coordinate and sound parameters of the actors. The coordinates for listeners and actors can be updated at any time.

3.2.1.2 Determining Pan and Surround Pan

The proper pan and surround pan values are determined for the sounds requested for the actors, based on the positional relationship between the listener and actors.

3.2.1.3 Volume Attenuation

Every time the distance exceeds the maximum sound volume range by one attenuation unit, the volume of sound requested for an actor is attenuated at the ratio specified by the attenuation rate. The attenuation rate and method (curve) are then set from SoundMaker for each sound.

Figure 3-3 Example When the Attenuation Unit Distance Is Set to 1000

3.2.1.4 3D Filter Control

A biquad filter is applied to the sound requested of the actor, in accordance with the distance between the listener and actor. The extent to which this biquad filter is applied is linked to the volume attenuation, so that distance is expressed by decreasing volume.

3.2.1.5 Voice Priority Attenuation

The voice priority of the sound requested for an actor is attenuated according to the actor-listener distance. In addition, the voice priority attenuation and the volume attenuation are proportional. For these reasons, the voice priority is attenuated by the same percentage as the volume and in relation to the distance-based priority maximum reduction value. Even if the actor-listener distance is so large that the sound becomes inaudible, the voice priority attenuation will not exceed maximum priority reduction.

3.2.1.6 Doppler Effect Expressions

The Doppler effect can be heard as a change in a sound's pitch according to the relative speed between the sound's source and its listener. This effect can be expressed by changing the pitch for the sound requested of an actor according to the speed at which the listener and actor are moving.

To change the pitch for the sound requested of an actor, the speed between the listener and actor is first calculated from the change in their positions since the last function call. Then this speed is used to apply the Doppler effect with both the speed of sound that was set in advance and the Doppler factor parameter.

The Doppler factor is a parameter that adjusts the level of the Doppler effect. SoundMaker specifies a Doppler factor for each sound.

3.2.1.7 Playback Switch Control

Set for each sound, the playback switches can be used to enable or disable the parameters controlled by the 3D sound system. Playback switch specifications for a sound are created using SoundMaker.

3.2.2 SoundMaker Operations

SoundMaker sets the operations when the sounds are requested as 3D sounds.

3.2.2.1 Specifying Playback Switches

You can set playback switches for volume, pan, surround pan, voice priority, and 3D filters for each sound you create. A playback switch sets the enable/disable control of each parameter that uses the 3D sound system. For example, you could specify that the 3D sound system determine pan and surround pan while the volume does not change, or you could set the voice priority such that the sound is always played, no matter how far away it is (that is, how low the volume is), and so forth.

3.2.2.2 Specifying the Sound Attenuation Curve

You can set the sound attenuation curve to be logarithmic or linear for each sound you create. Normally, the logarithmic curve generates a more realistic effect, while the linear curve works well for the sounds of a single actor, such as the pounding surf or a river.

3.2.2.3 Specifying the Volume Attenuation Rate

Specifying a volume attenuation rate for each created sound enables you to make the important sounds loud, even if they are far away, and to make unimportant sounds attenuate quickly.

3.2.2.4 Doppler Factor

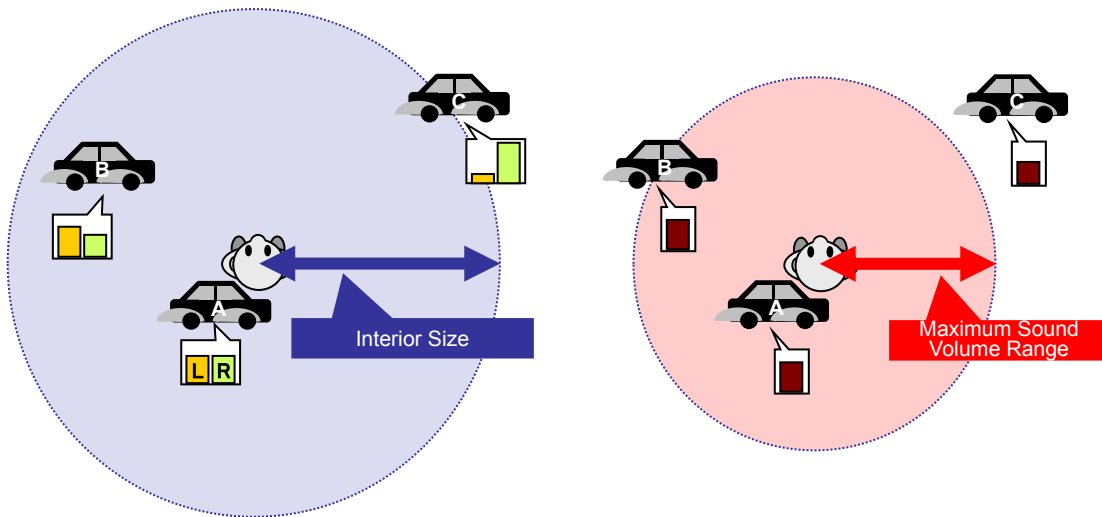
You can specify a Doppler factor for each sound that you create. The Doppler factor is a parameter that adjusts the level of the Doppler effect.

3.3 Interior Size and Maximum Sound Volume Range

This section describes the difference between the interior size and the maximum sound volume range.

3.3.1 Relationship Between the Interior Size and Maximum Sound Volume Range

While the interior size setting is used to adjust the output balance of each speaker based on the positional relationship between the listener and the actor, and the maximum sound volume range is used to adjust the output volume based on the positional relationship of the listener and actor, the two function completely independently with no effect on one another.

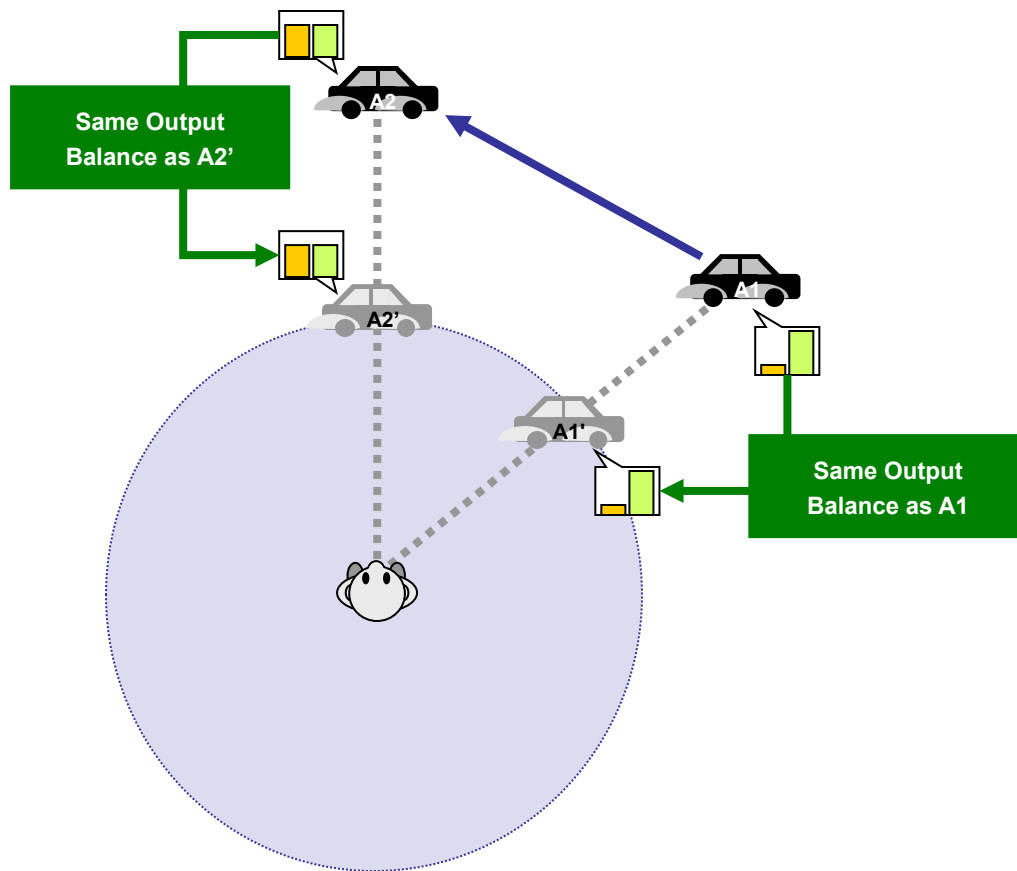
Figure 3-4 Interior Size and Maximum Sound Volume Range

3.3.2 Interior Size Properties

If the actor exists outside the circle surrounding the interior size, output balance is set so that it lies on the circumference when the shortest line to the actor is taken.

In Figure 3-5, output balance is set so that A1 is placed at position of A1' on the circumference, while A2 is placed at position A2' on the circumference.

Sound volume does not change due to changes in the interior size. The sound volume when A1 and A1' (A2 and A2') are located as shown is determined based on the maximum sound volume range.

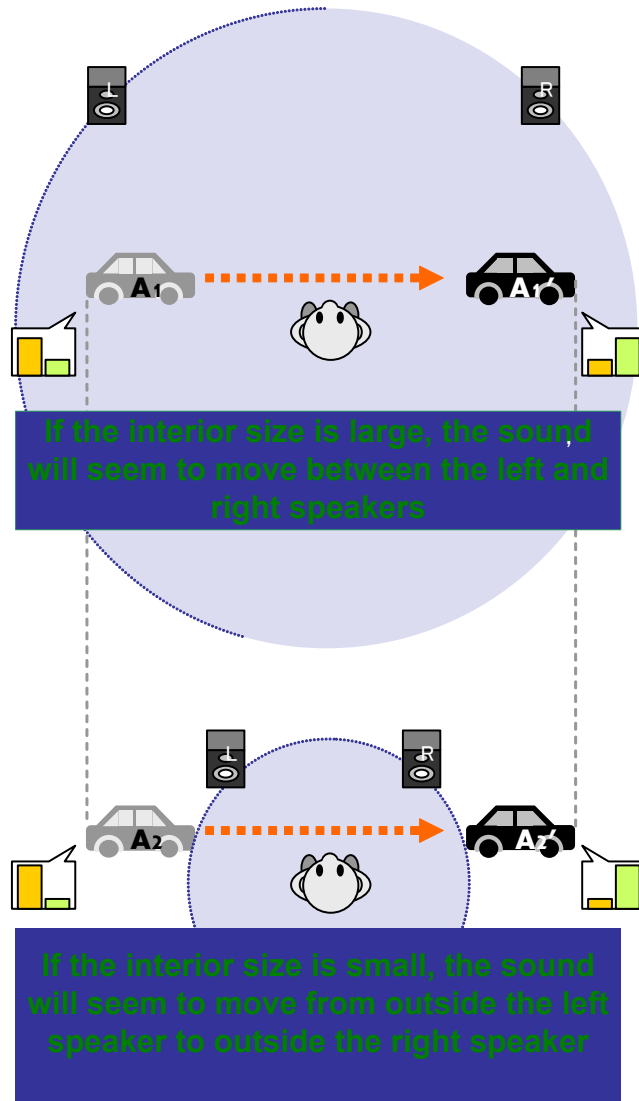
Figure 3-5 Interior Size Property 1

Because it is easier for the actor to be included in the circle when the interior size is large, sounds are easily collected in the center (do not easily move left or right), and output balance varies smoothly, even given the same speed of motion and distance of movement.

Conversely, because it is more difficult for the actor to be included in the circle when the interior size is small, sounds easily become broader (move easily to left and right), and the output balance can vary suddenly even given the same speed of motion and distance of movement.

Figure 3-6 Interior Size Property 2

Even if distance of movement and speed of motion are exactly the same...



4 Priority Behavior

4.1 Types of Priority

4.1.1 Player Priority

Player priority is a value that determines playback priority when sounds are played. A higher value has a higher priority.

4.1.2 Voicing Priority

Voicing priority is a value that determines audio-generation priority when various sounds generate audio for voices. A higher value has a higher priority.

4.2 Operation Specifications

4.2.1 Player Priority

4.2.1.1 Setting Player Priority

SoundMaker player priority (value of 0–127) is set for each sound. However, the program can also change this value.

4.2.1.2 Determining Player Priority

If the priority of Sound A, the sound to be played, is lower than that of Sound B, the sound whose priority is the lowest of all the sounds already being played, playback of Sound A will fail. If the priority of Sound A is higher than that of Sound B, playback of Sound A will succeed and that of Sound B will stop.

Player priority is determined in the following cases:

- The sound limit that is set for a player is exceeded and an attempt to play the sound is made.

The maximum number of sounds that can be played at the same time (the sound limit) can be set for each player. When an attempt is made to play a sound that exceeds this limit, player priority is assessed for sounds being played for the same player.

- An attempt is made to play a sound for which the maximum number set with the system has been exceeded.

The maximum number of sounds that can be played for stream sounds, for sequence sounds, and for wave sounds is set for the entire system. When an attempt is made to play a sound that exceeds this maximum for its type, player priority is assessed for all corresponding sounds being played.

- An attempt is made to use a sequence sound track or a stream sound channel that exceeds the maximum number set with the system.

The maximum number of sequence tracks and stream channels that can be played is set for the entire system. When an attempt is made to play sounds using a sequence track or stream channel that exceeds this value, player priority is assessed for all corresponding sounds being played.

However, because a single sound may require several tracks (channels), player priority assessment is repeated until all tracks (channels) can be allocated.

4.2.2 Voicing Priority

4.2.2.1 Setting Voicing Priority

Voicing priority is determined for each of the following types of sounds:

- Stream Sounds

Voicing priority is a characteristic, special value of the system that cannot be changed.

- Sequence Sounds

Voicing priority is set by the SoundMaker channel priority (value of 0–127). Priority of each track can be changed with the sequence track voicing priority (value of 0–127; default is 64), which can be set with a sequence command or MIDI control change. The final voicing priority value is calculated using the following expression:

(Final Voicing Priority) = (overall priority of the sequence) + (individual track priority – 64)

- Wave Sound Sets

Voicing priority is set by the SoundMaker channel priority (value of 0–127).

4.2.2.2 Determining Voicing Priority

Voicing priority is determined in the following cases:

- An attempt is made to produce audio for a voice that exceeds the maximum number of voices that can be produced simultaneously (default is 24).

When such an attempt is made, voicing priority is assessed. If the priority of Voice A, the voice to be played, is lower than that of Voice B, the voice whose priority is the lowest of all the voices already being played, playback of Voice A will fail. For all other cases, playback of Voice A will succeed and that of Voice B will stop.

- Voices are forcibly stopped due to DSP process overflow.

When DSP process overflow is generated, some voices are forcibly stopped. The voices that are already playing are stopped in the order from the voice that has the lowest voicing priority.

4.2.3 Determining Priority When Several Targets Are Assessed Under the Same Conditions

No feature exists in any priority processing to decrease the priority of a sound or voice target over time. However, when several sounds or voices have exactly the same priority value, earlier playback start equates to lower priority.

4.2.4 Special Priority Determination Processing

4.2.4.1 Voicing Priority During Release Phase

A special voicing priority that is the lowest of all voicing priorities is automatically applied to voices that have entered the release phase of the envelope. This operation can be suppressed by setting the option in SoundMaker for fixing priorities during release.

4.2.4.2 Voicing Priority for Stream Sounds

A special voicing priority that is the highest of all voicing priorities is automatically applied to stream sound voices. Except for determinations of player priority, voices that are being voiced in a stream sound are never forcibly stopped.

5 Pan Operations

5.1 Description of Terms

5.1.1 Pan Mode

Several types of pan operations, collectively referred to as pan mode, can be applied to stereo waveforms. The specification for pan mode is a setting that is only valid for stereo material.

5.1.2 Balance Pan

A type of pan mode, this is the name of operation that adjusts volume balance of the left and right stereo channels. The balance pan operation is performed on the stereo material the same way the normal pan operation is performed on monaural material; thus, there is no difference in volume between monaural and stereo sounds. Balance pan is suited for the material that has already been stereo-mixed for stream sounds.

5.1.3 Dual Pan

A type of pan mode, this operation adjusts the volumes of the left and right stereo channels as two independent monaural voices. For example, if the pan moves left, dual pan causes sound from the right channel to be also output to the left channel.

Dual pan is used with wave sounds on the stereo-sampled materials on which you wish to forcibly apply a pan.

5.1.4 Pan Curve

A type of volume attenuation curve, the pan curve is used to determine the volume of the left and right channels. The pan curve specification is valid for both stereo and monaural material, and the setting is also applied to the surround pan operation.

5.2 Operational Specification

5.2.1 Pan Operation on Stereo Waveforms

When a pan operation is conducted on stereo waveforms and the pan is placed at the center, the left channel is output to the left speaker, the right channel to the right speaker, and the sounds from each channel remain unmixed. This outcome is common to all pan modes if the pan is at the center; should the pan move left or right, the outcome depends on the pan mode. Sections 5.2.1.1 Balance Pan Operation and 5.2.1.2 Dual Pan Operation provide examples of operational differences caused by the pan moving to the right.

5.2.1.1 Balance Pan Operation

If the pan moves to the right, the left channel volume decreases, but the right channel volume either increases or maintains the volume at the center. Only the channel volume changes, not the position.

5.2.1.2 Dual Pan Operation

If the pan moves to the right, the left channel's pan moves to the right, but the right channel volume does not change. As a result, the left channel sound also emanates from the right channel.

5.2.2 Pan Mode Settings

In SoundMaker, pan mode settings are applied to the stream sound or wave sound level.

Table 5-1 List of Pan Modes That Can Be Set with SoundMaker

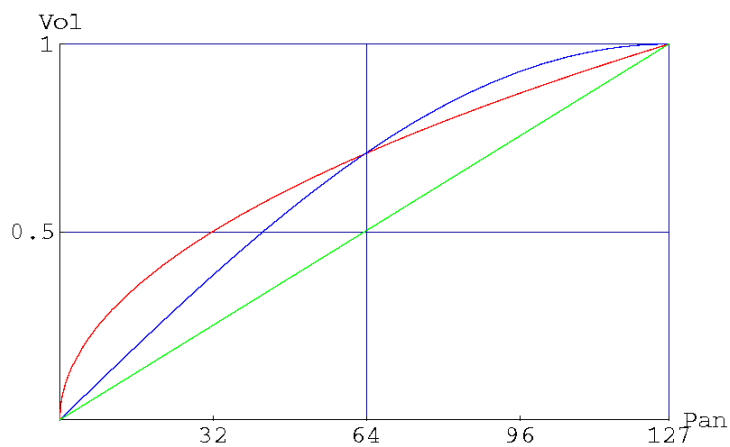
Pan Mode Setting	Operating Mode
Balance	Balance Pan
Dual	Dual Pan

5.2.3 Pan Curve Types, Pan Curve Settings

In SoundMaker, pan curve settings are applied to the stream sound or wave sound level. There are three choices for the curve type: square root, trigonometric function, and linear. It is possible to set on the curve the presence or absence of additional volume attenuation at the center. It is also possible to set a clamp that prevents the volume increase away from value at the center.

Table 5-2 List of Pan Curves That Can Be Set with SoundMaker

Pan Curve	Curve Type
Sqrt(-3dB)	Square Root, Center -3db
Sqrt(0dB)	Square Root, Center 0db
Sqrt(0dB,clamp)	Square Root, Clamp at Center 0db
Sin/Cos(-3dB)	Trigonometric Function, Center -3db
Sin/Cos(0dB)	Trigonometric Function, Center 0db
Sin/Cos(0dB,clamp)	Trigonometric Function, Clamp at Center 0db
Linear(-6dB)	Linear, Center -6db
Linear(0dB)	Linear, Center 0db
Linear(0dB,clamp)	Linear, Clamp at Center 0db

Figure 5-1 Pan Curve Types

Indicates the difference in each curve shape: Sqrt(-3db) is red, Sin/Cos(-3db) is blue, and Linear(-6db) is green.

6 Sound Parameter Control Operations

6.1 Setting and Controlling Sound Parameters

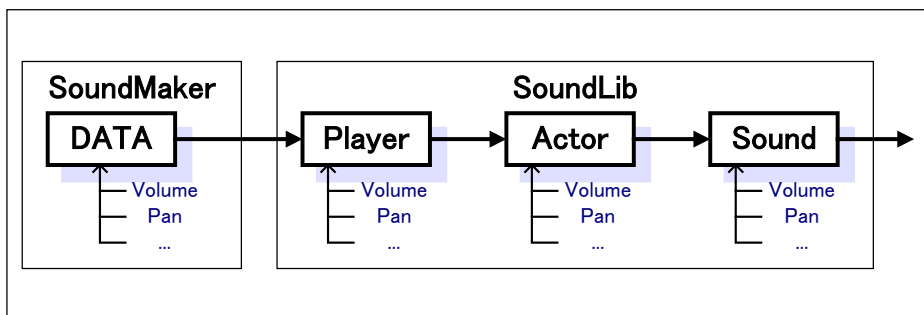
With SoundMaker, you can set various sound parameters, such as volume, pitch, and pan, separately for sequence sounds, stream sounds, and wave sound sets.

You can also use the `nw::snd` library to control each of the sound parameters per player, per actor, or per sound while the application is running.

This section explains the internal processing performed when the application is running and the `nw::snd` library is being used to control parameters of sounds created in SoundMaker by the sound designer.

In Figure 6-1, original values set for each sound in SoundMaker are indicated by “DATA”; values controlled by the `nw::snd` library are indicated by “Player” for player units, “Actor” for actor units, and “Sound” for sound units.

Figure 6-1 Sound Parameter Control



6.2 Operating Specifications

Each sound parameter is controlled by calculating a result using one of three methods (multiplication, addition, or priority) and applying the result to final playback of the sound.

The following explains the applicable parameter types and the calculation used for the three different methods.

6.2.1 Multiplication

This method determines the final parameter value by multiplying all of the parameters. Volume and pitch are the two parameters controlled using multiplication.

DATA x Player x Actor x Sound

6.2.2 Addition

This method determines the final parameter value by adding (or subtracting) all parameters around a midpoint of 0.0. Pan, sound pan, LPF cutoff frequency, main send, and effect send are the parameters controlled using addition.

With addition, each parameter's upper and lower limit values are predetermined. If a parameter exceeds the upper or lower limit during calculation, it is clamped.

DATA + Player + Actor + Sound

6.2.3 Priority

This method applies a priority ranking at each point in the internal processing flow where parameter control is desired (for example, a priority ranking at the player or sound level), and determines the final parameter value by overwriting the parameter according to the priority ranking.

biquad filter settings are controlled using priority.

The different levels of priority rankings are themselves ranked as shown in the expression below, with the levels on the left overriding the levels on the right.

Sound > Player > Actor > DATA

Table 6-1 Sound Parameter Types and Corresponding Control Methods

	SoundMaker (DATA)			nw::snd Library			
Parameter Type	STRM	WSD	SEQ ¹	Player	Actor	Sound	Method
Volume	○	○	○	○	○	○	Multiplication
Pitch	○	○	○	○	○	○	Multiplication
Pan	○	○	○	○	○	○	Addition
Sound Pan	○	○	○	○	○	○	Addition
LPF Cutoff Frequency	○	○	○	○	○	○	Addition
Main Send	○	○	○	○	○	○	Addition
Effect Send	○	○	○	○	○	○	Addition
biquad Filter Settings	○	○	○	○	○	○	Priority

¹ Sequence sounds are controlled using the same methods (volume is multiplied, pan is added, and so on) by referencing internal bank settings and several internal sequence settings.

7 Number of Sound Instances

The following values can be set within SoundMaker by selecting **Project > Project Settings** from them menu, and then selecting the **Sound Archive** tab in the Project Settings dialog box, which exposes the **Sound Archive Player** settings.

- Number of sequence sounds
- Number of sequence tracks
- Number of stream sounds
- Number of stream channels
- Number of wave sounds

This section describes the roles these parameters play when using the SND library.

7.1 Relationship to Memory Size

The five settings listed above are related to the number of instances (sequence sound instances, sequence track instances, etc.) included in an `nw::snd::SoundArchivePlayer` object linked to the sound archive (`.bfsar`).

As the number of each of the above increases, the more instances of each that can be maintained in the corresponding `SoundArchivePlayer` object, and the greater the number of types of sounds that can be played back simultaneously. However, the amount of memory returned by the `GetRequiredMemSize` and `GetRequiredStreamBufferSize` functions of the `SoundArchivePlayer` class also increases.

Specifically, the memory size returned by the `GetRequiredMemSize` function increases if any of the following are increased:

- Number of sequence sounds
- Number of sequence tracks
- Number of stream sounds
- Number of wave sounds

Similarly, the memory size returned by the `GetRequiredStreamBufferSize` function increases if the following increases:

- Number of stream channels

Note: Again, the values set for these five items influences the amount of memory required by the `SoundArchivePlayer` object. To keep the required amount of memory down, you must set values as low as possible in consideration of how likely it is that each type of sound will be played back simultaneously within the application.

7.2 Number of Sounds

In order to play each type of sound (stream sound, sequence sound, and wave sound), an instance of that sound class (stream sound, sequence sound, or wave sound) allocated inside the `SoundArchivePlayer` object is used. The numbers of sequence sounds, stream sounds, and wave sounds that can be allocated are limited by the settings made in `SoundMaker`. If your application attempts to play more than this number the following warning is displayed and the sound with the lowest priority is stopped.

```
snd_SoundInstanceManager.h:144 Warning:Sound (id:0XXXXXXXXX) is stopped for not  
enough seq sound instance.
```

7.2.1 Comparison with Player Sound Limits

The limit on player sounds is used to determine the maximum number of sounds of any type (stream sounds, sequence sounds, or wave sounds) that can be played. However, the “number of sounds” described previously can be used to determine the maximum number of sounds of each type that can be played.

For example, let's assume that the sound limit on the `PLAYER_ENEMY` sound player used for an enemy character is 8. Only eight sounds can be played simultaneously for the enemy character by `PLAYER_ENEMY` regardless of sound type.

Furthermore, the sound limit of the player is referenced when the player heap is allocated. An amount of memory equal to the sound limit times the heap size is allocated within the `SoundArchivePlayer::Initialize` function. This same amount of memory is reflected in the `SoundArchivePlayer::GetRequiredMemSize` function.

7.3 Number of Sequence Tracks

Sequence sounds are played using one or more sequence tracks. To check the number of sequence tracks used to play sequence sounds you can look at the arguments of the `alloctrack` command in the text sequence file.

As an example, here is the `opentrack_seq` sequence in `SampleData/Sound/Projects/simple/seq/se.fseq`.

```
opentrack_seq:  
    alloctrack TRACK_1 | TRACK_2 | TRACK_3
```

This tells us that a total of four tracks are allocated. (One track allocated without calling the `alloctrack` command plus three additional tracks (`TRACK_1` through `TRACK_3`) for a total of four tracks.)

If the allocation of a sequence sound instance succeeds, an attempt is made to allocate the required sequence track instances for that sequence. Memory only for the number of sequence tracks set by `SoundMaker` is allocated for the sequence track instances within the `SoundArchivePlayer` object.

If allocation of a sequence track instance fails, the warning shown below is displayed, and that

sequence sound stops. (Although the `StartSound` function itself returns success for that sequence sound, the sound stops playing shortly after this.)

```
snd_SequenceSoundPlayer.cpp:200 Warning:Failed to start sequence sound for not enough sequence track instance.
```

As an example, assume the following settings were made.

- Number of sequence sounds = 2
- Number of sequence tracks = 6

If an attempt is made to simultaneously play two instances of `opentrack_seq` (mentioned above), playback of the first succeeds, but allocation of the second sequence track instance fails because there are not enough stream channels. (Four tracks are required, but only two tracks remain because four tracks are already being used for the first playback.) Allocation of sequence track instances therefore fail, and the second sequence sound stops playing soon thereafter. Note, however, that the `StartSound` function returns success because allocation of sound instances has succeeded.

7.4 Number of Stream Channels

Behavior regarding the number of stream channels is similar to the behavior described above regarding the number of sequence tracks.

Stream sounds are played using one or more stream channels. The number of stream channels used for stream sounds is equal to the total number of channels for the waveform of each stream track set by SoundMaker.

For example, because `STRM_MARIOKART` accesses the stereo wave file named `kart_title.32.aiff` in `SampleData/Sound/Projects/simple/simple.fspj`, two stream channels are required. Three additional stream channels are required because `STRM_MULTI_TRACK` accesses three monaural waveforms (`BASS16.wav`, `DRUMS16.wav`, and `PIANO16.wav`).

If the allocation of stream sound instances succeeds for the stream sounds, an attempt is made to allocate the required stream channels for that stream sound. Memory for the number of stream channels set by SoundMaker is allocated for the stream channels in the `SoundArchivePlayer` object.

If allocation of stream channels fails, the warning below is displayed, and that stream sound stops. (Although the `StartSound` function itself returns success for that stream sound, the sound stops playing shortly after this.)

```
snd_StreamSoundPlayer.cpp:161 Warning:Failed to start stream sound for not enough stream channel instance.
```

As an example, assume the following settings were made.

- Number of stream sounds = 2
- Number of stream channels = 4

If an attempt is made to simultaneously play two instances of `STRM_MULTI_TRACK` (mentioned above), playback of the first succeeds, but allocation of the second sequence track instance fails because there are not enough stream channels. (Three channels are required, but only one channel remains because three channels are already being used for the first playback.) The second sequence sound stops playing soon thereafter. Note, however, that the `StartSound` function returns success because allocation of stream sound instances has succeeded.

8 Migration Guide

This chapter explains how to migrate from NintendoWare for Revolution, NITRO-System, and TWL-System to NintendoWare for CTR.

8.1 Migrating from NintendoWare for Revolution

8.1.1 Import Methods

To import an NW4R SoundMaker project, start NW4F SoundMaker and select **File > Import > Project from Other Platform**. Open the file selection dialog box and set **File Type** to “Project from other platform (*.rspj, *.sprj, *.sarc)” to specify an NW4R sound project.

You can import project (.rspj) files, sound-set (.rsst) files, bank (.rbnk) files, and text-sequence sound (.rseq) files. The original files are not changed when you import them; new files are saved using a different extension in the same folder.

8.1.2 Features Revised and Parameters Added/Removed When Importing

- Multiple pieces of wave data associated with a bank or a wave sound set are combined into a single “waveform archive.”
- The **Group Output** and **Waveform Sharing** items in the group list have been removed and the **Output Method** item has been added. This new item configures the output method to the sound archives for sound data registered with a group item. Its options are **Embed**, **Link**, **User Managed**, and **No Output**.
- You can now select **Multi-bank** for sequence sounds. Up to four banks can be set for a single sequence sound.
- **Register Related Items** has been added to group items. By changing this setting, you can include related items together in a group. For example, you can register a group of banks and waveform archives that are referenced by a sequence sound.
- You can now suppress string table output. Although NW4R SoundMaker forced string tables to be output to sound archives, you can now prevent this by opening the **Project Settings** and clearing the **Output the string table** check box on the **Sound Archive** tab in the Project Settings dialog box. This allows you to reduce the size of sound archives when sounds are not called using strings.
- Intermediate binary files are now always output to the cache folder. However, bank include files (.finl) and intermediate text sequence sounds (.fseq) generated by SMF are still output to the same location as the source files.
- The conversion process now generates HTML map files, rather than text files, for sound archives.
- Item IDs are now given to wave sound sets and sequence sound sets, allowing you to specify these items when you load data.
- Information that shows an item’s category is now embedded in the upper 8 bits of each ID for the various sound types (sequence sounds, wave sounds, and stream sounds), sequence sound sets, wave sound sets, banks, waveform archives, groups, and players.
- You can now convert sounds in parallel if they do not depend on each other. The number of sounds

that can be converted simultaneously is configurable from the **Options** screen.

8.1.3 Changes to the Sound Runtime

- Although wave sounds, sequence sounds, and stream sounds can be played on the PC-SDK, this feature should not be used (PC-SDK is no longer publicly available).
- A feature has been added to separately load the waveform archives associated with sequence sounds and wave sounds. See the reference manual for the `nw::snd::SoundDataManager::LoadData` function. Even if a bank or wave sound data is not loaded in advance, you can now load waveform archives associated with them.
- The `nw4r::snd::WavePlayer` class has been deprecated. But the same feature can be used with the Cafe-SDK's `nn::snd::Voice` class.

8.2 Migrating from NITRO-System and TWL-System

8.2.1 Import Methods

Before importing, you must first use NITRO-SoundMaker to convert the data because the bank ID and player ID references in the `@SEQ_TABLE` section of a MUS file refer to label (`.spdl`) files, which are generated after NITRO conversion and are required when importing.

Some of the NITRO-SoundMaker code is used to import projects, so projects that could not be loaded by NITRO-SoundMaker also cannot be imported by NW4F SoundMaker.

To import a NITRO-SoundMaker project, start NW4F SoundMaker and select **File > Import > Project from Other Platform**. Open the file selection dialog box and set **File Type** to “Project from other platform (`*.rspj`, `*.sprj`, `*.sarc`)” to specify a NITRO-SoundMaker sound project or a sound archive.

You can import project (`.spj`) files, sound archive (`.sarc`) files, bank (`.bnk`) files, text-sequence sound (`.smft`) files, and sequence-archive (`.mus`) files. The original files are not changed when you import them; new files are saved using a different extension in the same folder. Table 8-1 shows the relationship between the original and changed file extensions.

Table 8-1 Relationships Between the Original and Changed File Extensions

NITRO-SoundMaker	NW4F SoundMaker
<code>.sprj</code>	<code>.fspj</code>
<code>.sarc</code>	<code>.fsst</code>
<code>.bnk</code>	<code>.fbnk</code>
<code>.mus</code>	<code>.fsst</code> for the <code>@SEQ_TABLE</code> section of a sequence sound set <code>.fseq</code> for the <code>@SEQ_DATA</code> section
<code>.smft</code>	<code>.fseq</code>

These extensions are given as general examples and do not imply that other extensions cannot be imported.

8.2.2 Features Revised and Parameters Added/Removed When Importing

All data for items that are not in the NW4F SoundMaker GUI but were in NITRO-SoundMaker is deleted. For details on each of the items and features, see the User's Guide and the corresponding sections of this manual.

The following revisions have been made to each sound set node.

8.2.2.1 StreamSounds (STREAM in Legacy Projects)

The following items have been added to streaming sounds.

- Pan mode: The pan operating mode for a stereo waveform can be set to **Dual** or **Balance**.
- Pan curve: This configures the type of attenuation curve used to determine the left and right channel volume with pan operations.
- Multi-track streams: These allow you to set up to eight tracks for a single stream sound.

`MonoToStereoFlag` was removed.

8.2.2.2 Wave Sounds and Wave Sound Sets

Wave sounds and wave sound sets have been added. A *wave sound* loads waveform data into memory and plays it back. A *wave sound set* defines a combination of waveform data to load into memory at once.

8.2.2.3 Banks (BANK in Legacy Projects)

NW4F SoundMaker allows only one waveform archive related to a bank, even though four used to be allowed.

The first imported waveform archive is used and the second, third, and fourth are ignored.

8.2.2.4 Bank Tab (the Bank Window in Legacy Projects)

In a sample map, items that use a compressed type (`Type`) of `SWAV`, `PSG`, and `NOISE` are uniformly configured to ADPCM with an empty file path. An exclamation mark (!) is displayed as a precaution. Because the `SWAV`, `PSG`, and `NOISE` waveforms are not imported, you must prepare AIFF/WAV files for the bank instruments.

The volume, envelope hold time, pitch, and key groups were added.

Disable Envelope Release was replaced with the **Percussion Mode** parameter. The percussion mode is turned on for lists that were configured with the aforementioned **Disable** setting.

8.2.2.5 Sequence Sounds (SEQ in Legacy Projects)

The following items were added to sequence sounds.

- Playback start position
- Fixed priority on release

You can now configure **Multi-bank** for a sequence sound. Up to four banks can be set for a single sequence sound.

Imported banks are set to **Bank 0** in NW4F SoundMaker.

8.2.2.6 Sequence Sound Sets (SEQARC in Legacy Projects)

A sequence archive (SEQARC) is handled in the same way as a sequence sound set. It is displayed at the top of the tab screen. The result of loading a MUS file is displayed at the bottom of the tab screen.

You can register data with a sequence sound set just as you can with a sequence sound. However, a single sequence sound set can consolidate multiple sequence sounds.

Because you can also register individual sequence sound sets in a group, sequence data can be easily grouped using sequence sound sets.

8.2.2.7 Waveform Archives (WAVEARC in Legacy Projects)

Only separate loads are now permitted.

8.2.2.8 Players (PLAYER and STRM_PLAYER in Legacy Projects)

The PLAYER ChannelBitFlag data was deleted and STRM_PLAYER was combined with the player (PLAYER).

8.2.2.9 Groups (GROUP in Legacy Projects)

Items were added for the data size and output method.

Sound set items were added to the group items. The **Option** item was replaced with **Register Related Items**.

9 Troubleshooting

9.1 StartSound Succeeds but Sound Does Not Play

If the `nw::snd::SoundArchivePlayer::StartSound` function is successful but sounds do not play, this may be because the `nw::snd::SoundArchivePlayer::Update` function is not being called.

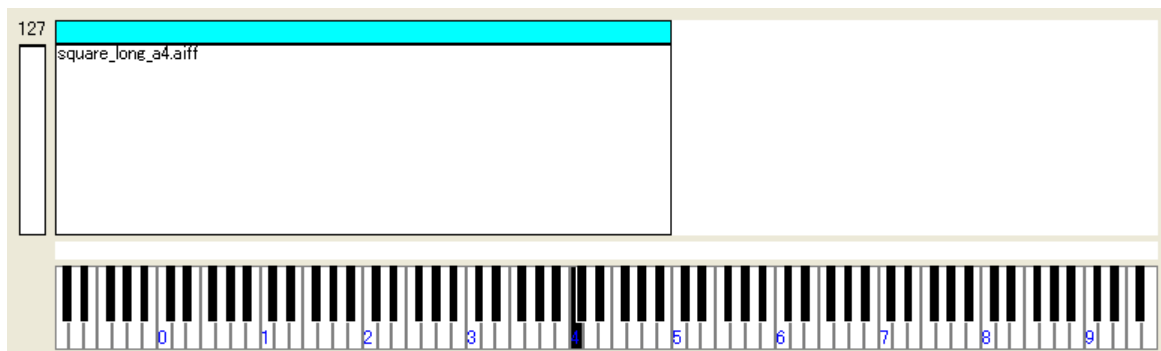
9.2 Certain Notes Do Not Play in Sequence Sounds

If a certain note does not play in a sequence sound, it may be because a waveform file has not been assigned to the region where that note should sound in the instrument sample map.

Check the instrument sample map in SoundMaker.

In the sample map example shown below, the `cn5` note does not play.

Figure 9-1 Region Where Waveform File Is Not Assigned



Revision History

Version	Revision Date	Class	Description
1.0.2	2015/07/03	Change	• 3.2 Operational Specifications Added to the description. • 3.2.1.4 3D Filter Control Corrected errors.
1.0.1	2015/01/19	Change	• 6.2.2, 6.2.3 Fixed typos.
1.0.0	2013/10/02	—	Initial version.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the U.S. and elsewhere.

Photoshop is a registered trademark of Adobe Systems Incorporated.

All other company and product names in this document are the trademarks or registered trademarks of their respective companies.

© 2013 Nintendo/HAL Laboratory, Inc.

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed, or loaned in whole or in part without the prior approval of Nintendo.